事件驱动、面向服务的物联网服务提供方法

乔秀全^{①*}, 章洋^①, 吴步丹^①, 程渤^①, 赵帅^①, 马华东^②, 陈俊亮^①

① 北京邮电大学网络与交换技术国家重点实验室, 北京 100876

② 北京邮电大学计算机学院智能通信软件与多媒体北京市重点实验室, 北京 100876

* 通信作者. E-mail: qiaoxq@bupt.edu.cn

收稿日期: 2013-04-01; 接受日期: 2013-08-23

国家重点基础研究发展计划 (973 计划)(批准号:2011CB302704, 2012CB315802)、国家自然科学基金 (批准号:61171102、61132001)、北京市科技新星计划 (批准号:2008B50) 资助项目以及北京市共建项目专项资助

摘要 物联网无缝集成了用户域、信息空间域和物理空间域, 实现了“人—机—物”的动态协同, 其应用常常涉及到跨业务域甚至跨组织的用户、业务流程和物理实体的实时动态交互. 因此, 物联网服务系统往往需要集成多个分布式的企业应用系统从而形成一个大规模复杂服务系统来实时协作完成一定的业务目标. 目前, 大多数已有研究工作主要集中于解决不同实体和应用系统间在动态交互环境下的互联和互操作问题. 然而, 如何在信息的提供方和消费者之间按需的分发感知信息、并基于物理世界所发生的变化动态的实时协同相关的企业服务或者业务流程做出快速反应尚没有系统深入地研究. 为此, 本文结合事件驱动架构和面向服务架构的设计思想, 提出了一种事件驱动、面向服务的物联网服务提供方法, 该方法能够便捷地支持感知信息的按需分发和聚合, 并实现事件驱动的跨业务域甚至跨组织的服务动态协同. 这种实现了“时间、空间和控制”解耦的服务提供方法极大地增强了物联网服务系统的灵活性和敏捷性, 允许系统及时地对物理世界的动态变化做出快速响应. 最后, 通过实际部署的区域集中供热监控及热计量物联网应用进行了验证.

关键词 物联网 服务提供 事件分发 服务协同 面向服务架构 事件驱动架构 复杂事件处理

1 引言

随着嵌入式计算和新型传感技术的高速发展, 信息与通信技术的目标已经从任何时间、任何地点连接任何人, 发展到连接任何物品的阶段. 人们希望将计算设施形成的虚拟世界和物理世界更加紧密地联系在一起, 使计算设备能够无缝地融入到人们的生活中, 通过实时地感知物理世界并进行动态交互, 从而提供更加智慧的普适服务. 在这一背景下, 物联网 (IOT, Internet of Things) 技术应运而生^[1], 大量的物理实体相互连接并集成到信息空间中, 通过网络来实现信息的提供和消费. 与现有的服务 (如电信服务或者互联网应用) 提供模式相比, 物联网服务提供呈现出一些新的特点. 具体而言, 现有的服务主要是解决“人—机”和“机—机”的交互, 也就是说现有的服务主要是面向一个二元问题域, 即用户域和信息空间域. 然而, 物联网服务还需要解决与物理世界的无缝集成和动态交互. 因此, 物联网服务本质上面临的是一个“用户、信息空间和物理空间”的三元问题域^[2]. 物理空间的引入使得环境感知成为了物联网服务的一个主要特征, 而这种实时的感知信息使得物联网服务系统始终处于

引用格式: 乔秀全, 章洋, 吴步丹, 等. 事件驱动、面向服务的物联网服务提供方法. 中国科学: 信息科学, 2013, 43: 1219–1243, doi: 10.1360/112013-113

一种动态变化的状态中. 物联网服务系统需要对大量传感器产生的海量感知信息进行实时汇聚、融合, 并将处理过的不同层次的信息按需地分发给不同的应用参与方, 从而及时地触发相应的业务系统进行协同. 因此, “环境感知 → 信息融合 → 智能决策 → 服务协同” 成为了物联网的服务提供模式, 物联网服务系统需要在大量分布式、多源感知信息的处理和融合的基础上, 快速地跨业务域甚至跨组织地协调人工、业务流程和物理实体进行交互. 由于所涉及的业务流程、人员和物理实体不仅类型多样, 而且物理世界在持续不断地变化, 导致物联网服务系统呈现出了实时动态变化的特点. 这种大规模、多源感知信息的采集、分发和融合, 以及服务系统需要对物理世界的动态变化做出相应地实时响应给物联网服务提供带来了新的技术挑战: (1) 如何跨业务域甚至跨组织地在多个分布式、松耦合的信息系统间按需分发、实时汇聚/融合和共享感知信息; (2) 如何基于实时感知信息实现大量异构服务、企业业务流程、物理实体和人员的有效协同来快速应对物理世界的动态变化.

目前, 物联网领域产业碎片化、建设重复化情况严重, 大量物联网应用系统在做低水平的重复开发, 系统重叠性高, 缺乏共性技术平台. 因此, 物联网服务提供机制引起了越来越多的重视和关注, 已有大量工作研究了通过移动代理或者中间件技术来实现物理世界和企业服务的集成^[3~5]. 然而, 在集成过程中面临着一个主要的障碍是大量已部署的传感器网络或者应用系统采用了专有业务平台和专有的技术, 这些非标准的接口协议导致应用系统与传感网络的集成非常复杂. 为了解决跨业务域的系统集成, 近几年, 研究人员将面向服务的计算技术 (SOA, service oriented architecture) 应用到物联网服务提供上, 用来解决服务和物理实体的异构性、互操作性和物联网服务的动态发现方面^[6~9]. 然而传统的 SOA 标准和技术的设计初衷主要是为了解决互联网环境下重量级的企业级服务的集成, 在解决物联网服务提供时存在一些局限^[10], 部分研究人员尝试提出了一些适合于资源受限的嵌入式设备的轻量级 Web 服务协议^[11,12]. 总体而言, 已有的研究工作主要集中在使用 SOA 技术或者更轻量级服务协议来实现不同物理实体和企业应用系统的互联互通和互操作性问题上. 事实上, 互联互通及互操作是物联网服务提供的一个前提条件, 但随之而来的另一个问题就是在解决了传感设备和系统的互联互通之后, 如何在一个分布式、松耦合的环境中实现感知信息在信息提供者和消费者之间的有效分发, 并基于物理世界发生的变化而动态地协调相关的企业服务流程做出快速响应. 目前, 大多数物联网应用系统是一种紧耦合、封闭式的服务提供模式, 也就是传感器网络采集相关的感知信息存储在一个集中的数据库中, 相关特定的应用系统通过访问数据库中的信息从而提供相应服务. 这种“客户端 – 服务器端”的系统结构极大的限制了系统的灵活性和扩展性, 不适合大规模分布式、复杂物联网服务系统的构建. 为了支持在不同的服务系统间自由便捷地交换信息并进行业务协同, 传统的以“请求 – 响应”为主的 SOA 架构经常用集中式的服务编排机制来解决. 然而, 这种紧耦合模式导致了信息的提供方和消费者彼此直接交互, 业务流程常常通过一些集中控制的流程编制语言如 BPEL (business process execution language) 或者工作流语言来进行描述. 这样当应用需求发生了变化, 如新集成了一个服务或者一个服务的退出, 这种紧耦合的、集中式的服务编排必须重新修改相应的业务流程, 缺乏足够的灵活性来做出动态适配. 随着物联网服务系统越来越庞大复杂, 所涉及的第三方系统或者企业遗留系统非常多, 这就要求物联网服务系统有很强的整合能力及对异构、动态变化环境的适应能力, 能够通过快速开发或者重用/整合现有资源来达到快速响应业务需求.

为此, 本文首先分析了物联网服务的特征, 通过集成事件驱动架构和面向服务架构的优势, 提出了一种事件驱动、面向服务的物联网服务提供方法, 实现了感知信息的按需分发和事件驱动的服务动态协同. 这种“时间 – 空间 – 控制”解耦的方法能够较好地适应动态变化的物联网应用环境. 总结起来, 本文的主要贡献为:

- 为了支持感知信息在信息提供方和消费者之间的有效分发, 提出了一种基于发布/订阅机制的

图 1 现有服务的二元问题域

Figure 1 Two-tuple problem domain of the existent service

统一消息空间来支持大规模分布式、松耦合环境下的感知信息的异步通信、按需分发;

- 为了从海量、琐碎的感知数据流中迅速识别出有价值的事件并实时动态地做出响应, 提出利用复杂事件处理引擎来即时过滤、关联和聚合多源的简单事件, 使物联网服务系统具有对物理世界和业务系统所发生的变化情况做出实时反应的能力;

- 提出了一种事件驱动、跨业务域的服务动态协同机制, 通过使用隐式的事件链来驱动跨域的多服务系统并发执行, 提升了服务的响应速度和系统的敏捷性;

- 通过一个已部署的区域集中供热监控及热计量物联网应用验证了所提方法的可行性.

本文第 2 节分析了物联网服务的新特征和涉及的技术挑战; 第 3 节提出了事件驱动、面向服务的物联网服务提供方法, 包括系统架构、关键技术和服务提供机制; 在第 4 节, 利用一个已部署的区域集中供热监控及热计量系统验证了所提方法的可行性; 最后总结了本文工作并指出了下一步的工作.

2 问题提出及面临的技术挑战

2.1 物联网服务新特征

为了更好的理解物联网服务的新特征, 本文首先分析一下现有的电信网或者互联网服务的特征. 如图 1 所示, 现有的服务提供主要面临的是用户域和信息空间域的交互问题, 基本上以“请求-应答”的服务模式来提供服务 (如用户拨打电话或者发送短信、用户浏览网页或者点击查看视频内容、或者点击网上购物等), 只有少量类型的服务通过服务器端的个性化配置来主动推送一些用户感兴趣的内容 (用户订购的每日手机报或者天气预报通知等). 因此, 我们归纳现有的服务主要为一个面向“用户和信息空间”的二元问题域, 主要解决“人-机”和“机-机”的交互, 在物理环境感知方面的特征体现的不是很强, 虽然个别业务或者应用有一些, 但不是这些服务的一个基本特征.

而物联网无缝的将物理空间融入了信息空间, 物理对象成为了企业业务流程中的一个重要参与方^[13], 物联网服务系统可以通过网络方便地与这些物理对象进行实时交互、查询或者改变它们的状态, 发出相关控制指令等. 这样, 物联网服务不仅需要与用户和信息空间资源交互, 还需要实时的感知物理世界并动态的协同人、机、物来做出应对. 因此, 物联网服务需要具有高度的智能性和自主性, 在大多数情况下, 系统自身能够基于观测到的物理情景和相应的策略来实时地执行对应的业务处置流程. 如图 2 所示, 与现有的电信服务或者互联网服务相比, 物联网服务主要解决“人-机-物”的动态协同, 物联网服务的系统边界发生了变化, 从原来的二元问题域 (即“用户、信息空间”) 转变为三元

图 2 物联网服务的三元问题域

Figure 2 Three-tuple problem domain of IoT service

问题域 (“用户、信息空间、物理空间”). 可以看出, 物联网的服务提供环境与现有的电信或者互联网服务相比发生了较大的变化.

本质上来说, 物联网具有三个基本特征: ①互联互通②透彻感知③智慧服务; 其中, 互联互通主要解决多种异构网络、异构终端和传感设备的互联互通, 为感知数据的大规模交换和传输奠定基础; 透彻的感知主要解决对物理空间域的相关环境/上下文信息的采集和分布式智能处理, 为上层应用提供对物理世界变化的有意义的高层认知信息; 服务的智慧性体现在物联网服务系统需要根据物理空间的状态变化等情况, 能够尽可能地自主决策从而动态适应这种实时变化的物理世界. 这三者的关系是, 互联互通是基础, 透彻感知是关键, 智慧服务是目的. 可以看出, 环境感知计算是物联网服务提供中非常关键的一项支撑技术, 通过对物理世界中多源的、分散的感知信息进行分布式的、分层次的协同处理, 产生更有意义的高级信息, 从而对服务所处环境形成一个全面、完整的认知. 物联网具有的这种特有的感知特征促使物联网服务系统从现有服务的“控制流 + 媒体流”模式转变为“控制流 + 媒体流 + 感知流”模式. 现有的电信网或者互联网服务基本上是基于控制流和媒体流来设计的, 控制流是指各种请求/响应信息、呼叫信令、管理命令等控制操作; 媒体流是指各种语音、数据和多媒体承载/内容流. 在新型的“环境感知 → 信息融合 → 智能决策 → 服务协同”的物联网环境下, 对服务所处的物理环境的感知是建立控制流和媒体流的重要条件. 由于物联网环境下, 各种物理传感器能够透彻感知到物理空间域的实时状态信息, 因此, 本文将物联网环境下多源上下文信息的采集、传输、融合处理所形成的信息流称之为“感知流”, 提出“控制流、媒体流和感知流”相结合的物联网研究思路 (如图 3 所示), “三流”的有机协同是提供智慧型物联网服务的基础.

从以上分析可知, 物联网服务提供的问题主要是在对物理环境实时感知和信息融合的基础上, 如何灵活有效的协同各种应用系统来提供智慧的服务. 现有的电信网或者互联网服务提供, 更强调控制流和媒体流的协调 (比如用户拨打一个电话, 先发起呼叫信令, 对方接听后呼叫发起信令交互完成 (控制流), 然后建立承载通道来传递双方的话音 (媒体流); 再比如用户在网页上点播一个视频片段, 发出业务请求信息之后 (控制流), 服务器将相关内容和数据返回给客户端 (媒体流)). 可以看出, 现有的电信网/互联网服务提供实际上缺乏对物理世界的实时感知、数据流的实时分析和汇聚, 并形成决策来协调相关业务系统的能力. 而信息的感知和处理并智慧地协同不同的服务满足用户需求是物联网的核心本质.

图 3 物联网环境中“感知流”概念的提出

Figure 3 The “Sensing flow” concept of IoT

2.2 新兴的热计量物联网应用场景

为了更好的说明上述问题, 本文以一种新兴的物联网应用 – 支持热计量的区域集中供热监控及信息服务为例来说明. 热计量的目的主要是确定生产或消耗的热能数量, 从而监督和计算热源设备 (锅炉或换热器等) 的效率或作为向用户收取用热费用的依据. 根据部署在锅炉、换热站、供热管道、用户驻地的各种传感器和智能仪表, 热计量供热系统能够根据用户的个性化供热需求和室外气候条件自动或半自动地调整供热策略, 如气候补偿器可以根据室外气候的温度变化, 用户设定的不同时间段的室内温度要求, 按照设定的曲线自动控制供水温度, 实现供热系统供水温度的气候补偿. 因此, 热计量对加强供热的科学管理, 节约能源, 具有十分重要的意义. 与现有的按住房面积计费的粗放式供热模式不同, 热计量能够支持按用户使用的热量来进行精确计费. 随着供热行业管理的精细化, 信息系统变得越来越复杂. 感知数据不仅会用到供热实时监控系统, 同时也会被其他业务域的应用系统用到, 如复杂事件分析系统、告警系统、计费系统、维修系统和客户关系管理系统等, 如用户通过登录客户管理系统门户或者手机客户端应用能够随时随地查看当前家里的供热情况以及每天、每月使用的热量汇总, 甚至可以通过手机或者 PC 来远程调节室温等; 计费系统根据住户热计量数据生成用户计费账单; 当用户投诉家里的暖气片不热的时候, 客服人员或者维修人员可以方便的查看该住户的实时的供热情况, 同时也能看到该用户是否按时缴费的情况; 另外, 供热作为一种社会公共服务, 政府管理部门如街道供热办也需要监督和保证所管区域的供热是否达到政府规定的最低室温标准. 可以看出, 热计量数据会被多个分布式的跨业务域甚至跨组织的信息系统所分享和使用, 但不同的信息系统可能对不同层次的数据感兴趣.

在传统的“请求—响应”为主的 SOA 交互模式下, 当感兴趣的事件发生之后, 数据感知/采集服务域将会执行一个显式的、集中式的业务流程来调用相关的 Web 服务将该事件通知给其他的相关信息系统, 如图 4(a) 所示. 在这种模式下, 信息的提供方明确的知道信息消费者并直接与之进行交互. 这种紧耦合的方式给系统的扩展和动态演化带来了隐患, 如图 4(b) 所示, 当新集成了一个维修和应急救援系统时, 必须修改原来的业务流程, 增加新系统的 Web 服务调用. 因此, 这种传统的以“请求—响应”为主的集中式服务编制模式难以满足高度复杂、动态和分布式的物联网应用场景.

另外, 支持热计量的区域集中供热系统包含了大量的感知信息和业务事件, 系统常常需要自动化

图 4 物联网服务系统中跨业务域的信息共享和服务协同

Figure 4 Information sharing and service coordination cross domain in IoT service system

的实时处理感知信息流并做出及时的决策来响应不断变化的场景, 如当热水管道发生了漏水, 管道压力、回水量、回水温度和住户家里温度都会发生一些异常的变化, 供热系统需要聚合这些多源的感知信息从而判断出发生的隐含问题. 基于这些判断, 系统会在实时监控系统上显示告警信息, 维修通知将会发送给报修及应急救援系统; 当维修工作不得不影响到住户供热时 (如要关掉热水管阀门), 客服系统需要发送相关通知短信给受影响的住户; 如果万一故障问题变得越来越严重, 这个紧急事件还需要上报给政府相关管理部门 (如供热办). 因此, 物联网应用常常涉及到实时处理多源的感知信息流, 并基于事件来跨业务域甚至跨部门的动态协同相关的业务系统.

2.3 物联网服务面临的技术挑战

从上面的分析可以看出, 现有的服务提供机制基本上没有充分的考虑大规模感知信息的分布式处理以及如何根据物理世界所发生的变化来跨业务域甚至跨组织的有效协调不同的企业服务流程. 因此, 如何在一个开放、分布式的计算环境中有效的按需分发、实时聚合相关感知信息, 并实现灵活、动态的异构服务系统间的协同是物联网服务提供面临的挑战, 具体为:

(1) 跨业务域甚至跨组织地在多个分布式、松耦合的信息系统间按需分发、实时汇聚和共享感知信息.

目前, 大多数的已有物联网应用系统采用的是一种紧耦合的系统结构, 也就是说感知信息的处理基本上局限在一个封闭的环境中. 本质上, 目前的物联网应用主要还是处于一种企业内部物联网系统阶段 (即 “Intranet” of Things), 而不是真正意义上的基于互联网的物联网阶段 (即 “Internet” of Things)^[14,15]. 在大多数情况下, 感知信息以一种集中式的数据库存储模式来处理, 基本上没有涉及到在大规模、分布式计算环境下感知信息在不同企业信息系统之间的分发和处理. 信息提供者和消费者常常是以一种 “请求 - 响应” 为主的紧耦合方式直接通信和交互. 随着物联网技术的不断发展, 物联网应用将逐步演进为在分布式计算环境下跨业务或者跨组织的集成多个异构系统的大规模复杂服务系统. 信息的提供方和消费者需要解耦, 甚至没必要一定知道对方的存在. 不管其他系统运行如何, 每个系统都可以彼此独立正常运行. 服务系统将变得越来越自治, “域内高度自治, 域间高效协同” 将成为物联网服务系统的显著特征. 另外, 物联网应用往往包含大量琐碎的感知信息, 但是大量的感知数

据却蕴含着丰富的语义信息. 如何实时的处理事件间的关联关系, 将没有具体意义的简单原子事件聚合为富含逻辑意义的复合事件, 从而高效的、准确的从这些感知数据中分析挖掘出具有商业价值的信息, 并触发相应的处理流程, 是物联网服务智能化、自动化的重要基石. 因此, 如何有序的按需分发感知信息, 为相对独立的企业服务系统建立信息传输、共享和集成机制对促进物联网的发展非常重要.

由于大规模物联网感知信息的动态分布式共享特征, 迫切需要一种能够解决异构、松耦合、跨业务域的信息交换和聚合机制, 使信息提供方和消费者在时间、空间和控制方面完全解耦. 因此, 需要一种支持异步事件通信的机制来满足物联网应用的开放协作、分布式动态交互的需求.

(2) 事件驱动的跨业务域或者跨组织的服务动态协同机制.

在物联网环境中, 用户、企业业务流程和物理实体对象之间的分布式、异构、动态、自主交互的特点要求一种动态可扩展的、开放式的服务提供机制来最大程度地降低系统间的耦合度, 实现多个异构服务系统之间的动态协同来快速应对物理世界发生的实时变化. 由于物联网应用往往是通过集成多个分布式的企业信息系统而形成的一个大规模复杂服务系统, 感知信息需要在多个异构信息系统间的分发和共享. 在传统的以“请求—响应”为主的面向服务架构下, 系统人员的主要精力集中在通过定义各个子系统的标准接口, 然后提供一个预先定义的集中式的业务流程描述来支持异构系统之间的交互. 然而, 大规模物联网服务系统所涉及的跨业务域甚至跨组织的信息系统常常具有高度的自治性, 各个系统往往由于技术或者管理原因并不情愿做出相应的改变, 这就导致要动态协调这些异构信息系统面临着重重困难. 传统的面向服务的业务提供平台一般通过一些预先定义的业务流程集中地协调服务, 前提假设是所有触发服务协同的事件和数据都已经明确定义在业务流程中. 实际上, 这种方式并没有考虑跨多个业务流程之间或者业务流程之外所发生的预想之外的事件. 这样, 在一种随机或者动态的服务交互模式中, 传统的面向服务架构并不能很好地解决一个复杂事件触发多个服务动态协同的问题, 而物联网的服务提供机制恰恰需要解决不同业务域之间的多个解耦系统之间的实时交互和协同. 因此, 如何解决跨业务域甚至跨组织的灵活的分布式服务协同, 是物联网服务提供面临的另一个技术挑战.

3 事件驱动、面向服务的物联网服务提供方法

任何技术或者架构思想都是由具体的业务需求驱动的. 面向服务架构 (SOA)^[16] 的出现是由于人们打破竖井应用并追求功能重用的强烈需求, 而事件驱动架构 (EDA, event driven architecture)^[17] 是一种设计和实现一个应用系统的方法学, 在这个系统里事件可传输于松散耦合的组件和服务之间, 通过引入异步的事件发布/订阅模式, 使得系统具备了快速响应业务事件的能力, 非常适合应用在不可预知的和异步的环境里. 然而, 由于物联网应用系统的复杂性, 具体体现为各种物理实体和信息系统的异构性、感知信息在多个信息系统之间的按需分发、多个服务系统之间的动态自主协同等特点. 单纯的 SOA 和 EDA, 都有各自的局限和适用的范围; SOA 主要从系统解耦的角度入手, 它侧重于将整个应用分解为一系列独立的服务, 并指定各种标准和基础设施来使得这些服务易于重用, 能够很容易地被各种平台上的应用来使用^[18~21]. 但由于传统的 SOA 主要适应于“请求—响应”的服务交互模式, 是一种同步的命令控制模式, 更多地是关注静态的信息, 所以不能很好地与动态业务匹配. 而 EDA 不是同步的命令控制模式, 恰恰相反, 它是一个异步的事件发布/订阅模式, 一个事件的产生可以触发一个或多个服务的并发执行. 当某个事件发生后, 系统的不同服务就能够自动地进行触发, 这样就把这些静态的功能通过隐式的事件流动态地协同起来, 使得系统具备了在松散耦合的分布式计算环境下实

图 5 事件驱动、面向服务机制

Figure 5 Event-driven SOA approach

时感知和快速响应事件的能力.

可以看出, 事件驱动架构对面向服务架构具有很好的补充作用, 相互独立的服务系统之间通过事件机制完成一定的服务协作, 非常符合物联网的“域内高度自治、域间高效协同”的特点, 既提高了对不断变化的业务需求的实时响应, 又最大限度地减少了对已有应用系统的影响, 可以更容易开发和维护大规模分布式应用程序和不可预知的服务或异步服务. 目前, 已经有一些集成 SOA 和 EDA 的研究, 如 Overbeek S, et al.^[22] 提出了一种灵活的、事件驱动、面向服务的电子政务系统架构; Dasgupta S, et al.^[23] 提出了一种将用户活动和服务作为事件的事件驱动的服务组合模型. 然而这些工作主要是解决二元问题域 (也就是用户和信息空间) 的交互和协同问题, 并不是解决物联网的服务提供问题. 所以, 这些工作中所指的事件主要是来自业务系统 (如订单系统、仓储系统、或者电子政务系统等) 的业务事件, 而不是来自于实时监控的物理世界的实时数据流. 另外, 这些工作没有涉及从海量数据流中迅速检测出有意义的事件并做出及时响应. 因此, 在考虑物联网服务系统的异构性和动态变化的特征情况下, 课题组提出了一种物联网环境下事件驱动面向服务的协调交互式业务流程的初步设想^[24], 但该文主要侧重于通过事件来驱动不同的业务流程, 但事件如何分发、路由, 简单事件如何聚合为高层的复杂事件, 物联网资源如何来建模等问题都没有涉及, 而且该文尚没有全面、深入地阐述物联网服务的提供机制. 在此工作基础上, 本文全面、系统地提出一种事件驱动、面向服务的物联网服务提供方法 (EDSOA, event-driven service-oriented architecture). 该方法无缝的集成了 SOA 和 EDA 的各自优势, SOA 技术用来解决大规模分布式物联网环境中大量异构服务和物理实体之间的互操作和应用重用性问题; EDA 技术用来解决跨业务域甚至跨组织的感知信息的按需分发以及事件驱动的服务动态协同问题. SOA 和 EDA 的融合增强了系统的灵活性, 允许事件在解耦的系统和 service 间进行按需的分发, 通过即时过滤、聚集和关联事件的功能, 以极快的速度从海量数据流中迅速识别出有价值的事件 (包括可能对企业造成威胁或为企业提供商业机会的事件或模式) 并实时动态地做出响应, 为企业提供了响应这些实时动态业务所需的能力, 这对具有高度自动化要求的物联网服务系统来说非常适合.

与传统的 SOA 不同, EDSOA 更适合于动态、异构的物联网服务环境 (如图 5 所示), 事件分发网络降低了不同服务系统的耦合度, 这种解耦的系统或业务流程具有最少的直接交互, 实现了“时间、空间和控制”的解耦, 时间解耦主要是指参与交互的各方不需要在同一时间参与交互, 也就是说是一种异步的通信交互模式; 空间解耦是指参与交互的各方彼此不需要知道对方; 而控制流解耦就是在这种基于发布/订阅的事件驱动的服务协同中, 没有一个统一的集中式的服务编排流程 (如 BPEL 或者工作流流程) 来显式地描述整个系统流程, 而完全是一种松耦合的、隐式的事件流驱动的方式. 也就是说

图 6 事件驱动、面向服务的物联网服务提供架构

Figure 6 EDSOA-based IoTs services provisioning architecture

在这种模式下,一个新系统的加入或者一个旧系统的退出或者修改对其他系统没有任何的影响.通过发布/订阅的机制,事件分发网络能够支持感知信息或者业务事件在多个分布式、松耦合的信息系统间的按需分发和共享.而且 EDSOA 支持一对多或者多对多的异步事件通信模式,增强了服务的并发执行能力.通过事件的发布/订阅机制,可以用一种隐式的事件链来驱动不同的服务系统的动态协同.通过瞬时的过滤、聚集和关联事件,EDSOA 可以快速地检测出事件类型,从而帮助企业快速、恰当地响应和处理这些事件.因此,针对物联网服务环境的特点,可以采用 EDSOA 技术来构建信息的共享交换平台,实现跨部门、跨平台、跨应用系统的感知信息和业务事件的共享与交换,并对跨业务域甚至跨组织的服务协同提供支撑和保障.

基于以上的分析和考虑,本文提出了一种基于 EDSOA 的物联网服务提供架构(如图 6 所示).接入代理从传感器或传感器应用系统获得原始的传感数据,并使用资源描述模型转换异构的或专有信息为统一事件.然后,接入代理调用事件代理的发布/订阅接口,发布事件到基于发布/订阅机制的统一消息空间.统一消息空间负责事件的发布、订阅和路由.统一消息空间中发布的事件可以被其他相关的物联网服务系统订阅.除了感知信息,物联网服务系统也可以使用统一消息空间来将控制指令作为事件发布,从而发送到相关的传感器或者控制设备.一旦订阅的事件发生,统一消息空间将通知所有的相关订阅者.同样,物联网应用系统也可以发布/订阅应用层的业务事件(即由应用系统产生的事件,如一个审批流程的结束,或者一个新订单的生成),以实现不同应用系统间的服务协同.例如,一旦复杂事件处理系统检测到紧急报警事件,该通知将被发送到实时监控系统的可视化界面显示报警信号提醒值班人员,同时该报警事件也会被发送到 workflows 业务系统来触发相关的报警处置流程来处理此事件.可以看出,不同的服务系统彼此间不再直接交互,而是通过发布/订阅机制完成事件的异步通信和服务的动态协同.统一消息空间是该架构的核心部分,进一步解耦了不同的物联网应用系统.

构成系统架构的主要组件如下:

(1) 基于发布/订阅的统一消息空间.

统一消息空间利用发布/订阅机制来分发事件,其主要部件是提供事件发布、订阅、通知和路由功

图 7 事件代理的结构

Figure 7 Framework of event broker

能的事件代理。事件的发布者和订阅者基于订阅列表, 使用事件代理彼此通信。每个订阅者发送一个或多个订阅请求到事件代理, 标识其感兴趣的事件类型。当事件发生时, 事件代理通过路由机制将该事件交付给订阅了该事件的所有订阅者。事件代理通常执行存储和转发功能, 将消息从发布者路由到订阅者。如图 7 所示, 事件代理由三部分组成: 发布/订阅接口、事件订阅表和核心应用功能。发布/订阅接口为事件发布者和订阅者提供使用统一消息空间的接口服务。在本文中, 统一消息空间采用基于主题的发布/订阅机制。基于主题的系统, 事件被发布到“主题”上, 订阅者将接收发布到订阅主题的所有事件, 同一主题的所有订阅者将收到相同的事件。订阅者或发布者首先需要注册到事件代理上。然后, 它可以作为发布者发布事件, 作为订阅者订阅已有的事件, 或同时作为发布者和订阅者。为了支持事件通知, 事件代理通过 Web 服务通知 (WSN) 规范¹⁾ 实现了事件推送功能。此外, 事件代理还提供了主题退订接口, 以便订阅者也可以退订相关主题。为了支持事件路由, 事件代理维护了两种信息: 事件集和主题订阅表。事件集存储有关主题的最新发布的事件。事件路由模块根据主题订阅表处理接收到的事件。如果订阅者正好由此事件代理管理, 则该事件将通过主题通知接口直接通知给订阅者。当用户未在该事件代理注册时, 该事件将被路由到相关代理。为了保持统一消息空间中事件订阅的一致性, 每个事件代理需要根据下面的事件路由策略实时同步统一信息空间的订阅信息。

为了支持大规模分布式事件分发, 事件代理可以形成不同的簇, 以提供簇级的可扩展性, 而成簇的代理连接在一起扩展了地域范围。因此, 统一消息空间形成了如图 8 所示的由一组事件代理构成的一个分布式的覆盖网络。在这种分簇拓扑结构中, 统一消息空间需要支持簇内路由和簇间路由。每个簇中, 一个特定事件代理负责簇间路由, 称之为委托代理。发布者或订阅者连接到统一消息空间中的一个事件代理, 并通过该代理发布或订阅感兴趣的事件。当一个代理收到其客户端的订阅时, 该代理转发订阅到簇内邻居节点, 而本簇的委托代理负责转发订阅信息到其他簇。同样, 当一个代理从其客户端收到发布的事件, 通过统一消息空间转发事件到匹配订阅的代理。然后, 这些代理交付事件给感兴趣的订阅者。物联网服务可以将统一消息空间作为事件池, 通过发布/订阅接口接入和访问。

统一消息空间的事件基于订阅信息表来进行分发和路由。订阅表中的每一项由两部分组成: 订阅主题和订阅者通信地址。普通的事件代理维护两种订阅表: Registered_clients_subtable 和 Neighbor_hosts_subtable, Registered_clients_subtable 存储直接注册到此代理的客户端订购信息, Neighbor_hosts_subtable

1) OASIS. Web Services Notification (WSN). <http://www.oasis-open.org/committees/wsn/>

图 8 基于分簇拓扑结构的统一消息空间

Figure 8 Cluster-based unified message space

图 9 基于主题的事件订阅机制

Figure 9 Topic-based subscription mechanisms

存储簇内其他事件代理的订阅信息. 与普通代理相比, 由于簇的委托代理负责簇间路由, 还需要维护一个 Groups_subtable, 用以存储其他簇感兴趣的订阅信息. 图 9 显示了当簇 1 的订阅者 1 订阅了主题 X 且簇 2 的订阅者 2 订阅了主题 Y 时, 订阅表发生的变化. 因此, 当代理接收到其注册用户发布的事件时, 将查找订阅表并转发该事件到相关订阅者. 事件首先在簇内路由, 如果其他簇也订阅了该事件, 则继续簇间路由. 关于统一消息空间这个覆盖网络中事件代理节点的负载均衡优化问题的解决方法具体参见 [25].

图 10 资源描述语义模型中本体的组织层次

Figure 10 Ontologies level used in the semantic model

(2) 资源描述模型.

物联网服务提供中一个重要的特征就是如何在大量异构的业务系统之间对物理世界的感知信息进行分发、聚合和语义共享, 其中涉及到的一个问题就是事件的语义互操作问题, 通过这种“时间、空间和控制”解耦的物联网服务系统, 各个系统之间进行协同和交互的一个共同基础就是事件, 因此, 需要对事件进行标准化的建模, 而对事件的建模离不开对物联网资源的建模, 涉及到各种领域概念和术语或者通用的一些概念. 为此, 我们采用资源描述模型来解决此问题, 接入代理从传感器或传感器应用系统获得原始传感数据, 并使用资源描述模型^[26]将异构或专有信息转换为统一事件. 资源模型使用链接数据并扩展了现有本体, 如 W3C SSN(Semantic Sensor Network)²⁾等. 该资源模型可以表达资源、上下文信息和领域知识. 基于资源模型, 接入代理可以自动获取资源, 生成基于本体的语义上下文信息和事件.

图 10 显示了资源描述语义模型中使用的本体组织层次. 顶层 DOLCE 是 W3C SSN 采用的通用本体. 上层主要基于 W3C SSN 来描述传感器资源和观测数据. 由于 SSN 本体主要集中在描述传感器设备、观测数据和平台方面信息, 所以需要在实体模型和领域知识等方面对它扩展. 使用 OWL-Time 来表达时间信息, 参考了 SemSOS O & M-OWL^[27]的信息模型和观测数据模型. 使用 Linked Data 技术与 FOAF(Friend of a Friend)³⁾, GeoNames⁴⁾, Linked-GeoData⁵⁾和 DBpedia⁶⁾等本体关联. FOAF 是描述社交网络中的人和关系的模型, 本文将情境模型中的人员模型与 FOAF 关联可以增加模型的通用性获取更多的附加信息. GeoNames 和 Linked-GeoData 是一套全球范围的地理位置本体, 通过将资源模型与他们关联可以表达实体或资源的全局位置. DBpedia 是从维基百科抽取出的结构化数据, 提供了很多通用概念和概念间的关系. 底层领域本体对上述的本体进行扩展, 使模型能够表达领域知识.

2) W3C Semantic Sensor Network Incubator Group Semantic Sensor Network Ontology. <http://www.w3.org/2005/Incubator/ssn/ssnx/ssn>

3) Brickley D, Miller L. FOAF Vocabulary Specification, [2013-03-23]. <http://xmlns.com/foaf/spec/>

4) GeoNames Ontology. http://www.geonames.org/ontology/ontology_v3.1.rdf

5) LinkedGeoData Data Set. <http://linkedgeodata.org/Datasets>

6) DBpedia Ontology. <http://dbpedia.org/Ontology>

```

<ObjectPropertyAssertion>
  <ObjectProperty IRI="#entity"/>
  <NamedIndividual IRI="#householdTempLow"/>
  <NamedIndividual IRI="#household_3"/>
</ObjectPropertyAssertion>
<ObjectPropertyAssertion>
  <ObjectProperty IRI="#resource"/>
  <NamedIndividual IRI="#householdTempLow"/>
  <NamedIndividual IRI="#householdTemp_sensor1"/>
</ObjectPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#alarm_form"/>
  <NamedIndividual IRI="#householdTempLow"/>
  <Literal datatypeIRI="&xsd:string">light</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#alarm_form"/>
  <NamedIndividual IRI="#householdTempLow"/>
  <Literal datatypeIRI="&xsd:string">voice</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#alarm_level"/>
  <NamedIndividual IRI="#householdTempLow"/>
  <Literal datatypeIRI="&xsd:short">2</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#alarm_time"/>
  <NamedIndividual IRI="#householdTempLow"/>
  <Literal datatypeIRI="&xsd:dateTime">2012-10-12T23:05:04</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#threshold_alarm"/>
  <NamedIndividual IRI="#householdTempLow"/>
  <Literal datatypeIRI="&xsd:double">18</Literal>
</DataPropertyAssertion>
<DataPropertyAssertion>
  <DataProperty IRI="#value"/>
  <NamedIndividual IRI="#ch4Alarm"/>
  <Literal datatypeIRI="&xsd:double">1.1</Literal>
</DataPropertyAssertion>

```

图 11 基本观测事件的片段

Figure 11 Fragment of simple observation event

资源模型包括资源描述和实体描述两部分。资源描述部分用于描述和动态维护传感器、激励器资源的信息。实体描述用于描述被监控的对象和对业务场景中的情景信息进行建模。在本文的语义模型中, 资源模型中的 observation 属性与实体模型中的领域属性关联。我们称这种关联关系为资源实体绑定。绑定用于将观测数据提升为情景信息。绑定关系可能是静态的或动态的, 同时, 可能是自动生成的或者人工配置的。当需要产生绑定关系时, 需要综合考虑资源类型、资源 tag、资源观测类型、信息质量这几个方面的信息。

接入代理根据资源描述把异构的原始传感信息转化为统一的观测信息, 并产生简单的观测事件。资源实体绑定用于将观测信息提升为情景信息; 基于领域知识、情景信息与其他已有信息关联形成语义情景信息。图 11 显示了一个基础观测事件的本体片段, “entity” 属性表示此事件所对应的实体, “resource” 属性表示产生此事件的资源。此外, 事件内容还包括报警级别、报警时间、观测数值、数据单位、阈值等信息。

图 12 接入代理架构

Figure 12 Architecture of access agent

图 13 告警事件

Figure 13 Alarming event

(3) 接入代理.

接入代理实际上是一种发布者/订阅者和统一消息空间之间的消息适配器. 一方面, 它需要与各种传感器系统或应用系统进行通信, 以采集感知信息, 或通过一些专有接口传递控制动作. 另一方面, 它也需要通过发布/订阅接口发布或订阅感兴趣的事件. 接入代理的主要结构如图 12 所示. 数据采集是预处理和采集数据的过程. 数据采集管理为用户提供了一个人机界面以配置采集参数, 用户可以轻松地添加或删除相关的采集参数, 以适应应用需求的变化. 资源映射模块主要实现数据转换任务. 接入代理从传感器或业务系统接收数据后, 根据资源描述模型描述信息. 这样, 接入代理可以屏蔽数据的异构性, 为统一消息空间提供语义互操作的基础. 同时, 接入代理也可以作为订阅者订阅感兴趣的事件. 因此, 接入代理需要以异步方式处理事件通知. 事件通知处理模块为注册的事件代理提供服务调用接口. 一旦事件代理收到订阅的事件, 它会通过注册的通知接口通知接入代理. 此外, 用户还可以使用代理来处理一些简单事件, 这些事件与具体的、可测量的变化条件直接相关. 利用该机制, 用户可以按需配置外部的告警信息. 例如, 用户可以自定义温度阈值, 当观测值超过阈值时, 将生成如图 13 所示的特殊告警事件, 然后发布到统一消息空间. 报警处理系统和实时监控系统将收到告警事件. 一般来说, 由于大多数的传感器系统和业务系统使用专用的数据格式, 需要在系统中自定义代码封装和解析发布/订阅事件, 以将这些系统集成到统一消息空间中.

```

select a.BoilerRoomId, a.BackWaterPressure, b.BackWaterTemperature
from pattern [every a=BackWaterPressureSensor ->
    b=BackWaterTemperatureSensor(BoilerRoomId = a.BoilerRoomId) where timer:within(1 min)].win:time(2 hour)
where a.Status=' warning' and b.Status=' warning'
group by a.BoilerRoomId

```

图 14 事件模式的声明

Figure 14 A statement of event pattern

(4) 复杂事件处理.

从上面的描述中可以看出, 接入代理解决了信息采集和预处理问题, 统一消息空间提供了信息提供者和消费者之间灵活的事件分发机制. 事实上, 物联网应用往往包含大量感知信息和业务事件, 实时或准实时的物联网服务系统要求从潜在的多源事件流中抽取隐含的高层含义 (如高层上下文信息或潜在的商业机会), 并做出低延时的、智能决策从而对变化的情况做出合适的实时反应. 因此, 面向实时数据流的复杂事件分析, 在物联网应用中发挥着越来越重要的作用. 复杂事件处理将系统数据看作不同类型的事件, 通过分析事件间的关系, 建立不同的事件关系序列库, 利用过滤、关联、聚合等技术, 最终由简单事件产生高级事件来驱动业务流程. 复杂事件处理 (CEP, complex event processing) 是物联网应用智能化的关键支撑技术. 为了支持实时事件流处理, CEP 引擎采用的是存储事件查询语句和实时过滤数据的模式, 而不是一般的存储数据并对存储的数据运行查询的模式. 输入/输出事件流适配器负责适配不同的事件格式, 如 XML 或 Java 对象. 为了支持复杂事件检测规则的定义, 采用专门的事件处理语言 (EPL, event processing language) 来表达丰富的事件条件、事件的关联和跨越的时间窗口. EPL 是一种类 SQL 语言, 拥有 SELECT, FROM, WHERE, GROUP BY, ORDER BY 子句. 逻辑运算符, 如 AND, OR, NOT, 可用于复杂事件的表达. 不同事件的时间序列可以用 “ $\rightarrow$ ” 运算符表示. 例如, $A \rightarrow B$ 表示事件 A 发生后, 接着事件 B 也发生了的情况. 对于时间相关的事件流处理, 与时间相关的原语, 如 timer: interval, timer: at, timer: within, 可以用在 where-conditions 中以控制子表达式的生命周期. 基于这些操作符, 用户可以定义感兴趣的事件检测规则 (即声明). 通常, EPL 支持模式匹配、任意条件下的事件联合以及创建时间窗口等. 图 14 显示了一个以供热监控系统为例的事件模式声明, 其含义是以 2 小时为一个时间段, 将一个锅炉房中在 1 分钟之内回水压力和回水温度先后都出现告警状态的事件检测出来, 并按锅炉房 id 对事件查询结果进行分组.

CEP 引擎的结构如图 15 所示. 为了方便事件检测规则或者事件模式的定义, CEP 引擎提供了一个可视化的事件模式编辑器. 用户可以自定义规则和条件 (称为模式) 对感兴趣的事件进行过滤. 生成的事件检测规则 (即声明) 运行时加载到 CEP 引擎. 然后, 事件模式检测模块实时聚合来自分布式系统的事件信息并应用规则来识别模式. 一旦检测到预定义的模式, 事件触发通知将发送到相关的事件监听器, 事件监听器可以根据具体应用需求决定随后的操作. 通过对简单事件的聚合而得出的高层事件也可以作为新事件再次发布到统一消息空间, 从而支持事件的嵌套级联处理模式. CEP 引擎有能力处理多个事件流, 识别有意义的事件. 此外, 为了方便自定义事件类型的处理, 用户可以通过可视化界面注册事件类型到 CEP 引擎, 以简化 EPL 声明的编写.

(5) 应用服务子系统.

由于物联网应用需要无缝协调 “人 — 机 — 物” 的动态交互, 因此, 物联网服务系统常常需要集成多个分布式、跨业务域的企业应用系统从而形成一个大規模复杂服务系统来实时协作完成一定的业务目标, 如 2.2 节基于热计量的区域集中供热监控及信息服务应用所述, 需要集成一系列的企业应用服务系统来完成特定的业务功能, 如感知数据呈现及监控系统、报修服务系统、客户管理系统、计

图 15 复杂事件处理

Figure 15 Complex events processing

费系统、报警及处置服务系统、商业智能分析系统、报表服务系统等。

(6) 事件驱动、面向服务的跨域/跨组织的服务动态协同机制。

通过感知、按需分发、过滤和聚合多源感知信息, 复杂事件处理系统可以检测出一些高层事件, 如火灾报警或异常货物移动。一旦感兴趣的事件被触发, 需要执行相关业务流程对变化的情况做出及时反应。考虑到全自动的服务动态组合技术目前并不现实, 一种灵活的、实用的服务协同机制是有必要的, 以满足应用需求的时变性和多种异构服务系统的动态协同特点。本文采取了两项措施来增强系统的灵活适配能力。

一方面, 功能相对独立的不同业务域的服务系统, 利用事件驱动的方式尽可能协作。这意味着, 业务系统主要使用事件的发布/订阅机制实现信息的交互。在这种方式中, 没有显式的、集中的服务编排流程来定义 Web 服务或任何其他服务的调用顺序, 而是采用异步的、隐式的事件链来驱动不同业务系统的动态协同。如图 16 所示, 业务系统 1 生成并发布事件 1, 而事件 1 被业务系统 2 和业务系统 3 订阅。当业务系统 2 和 3 收到事件 1 时, 执行相应的业务流程。业务系统 2 生成的事件 2 将进一步触发业务系统 3 的流程继续执行。这样, 当有新的需求变化时, 用户可以通过调整事件链的方式来改变业务处理流程, 即改变事件的发布和订阅。例如, 当一个新的业务系统集成到本系统并对事件 1 感兴趣时, 它只需要订阅事件 1, 而这不会影响现有的任何其他业务系统。事件的发布者不关心事件订阅者的存在。很明显, 事件驱动的服务协同本质上是异步的一对多、多对多的事件通信范式。因此, 这种事件驱动的服务协同机制有助于降低跨业务域甚至跨组织的业务系统的耦合度, 形成一个更灵活、更敏捷的物联网服务提供系统。

另一方面, 并不是说所有请求驱动的服务交互都应被事件驱动的服务交互全部替换。对于实现一个具体的业务系统, 仍然鼓励采用传统的请求驱动的、面向服务的技术, 以进一步提高系统的自身的灵活性和功能重用性。由于当前物联网应用系统缺少可重用的服务, 大部分现有的物联网系统的实现还是硬编码。为了提高业务系统的敏捷性, 一些明确定义的业务功能应该作为共享软件组件 (离散的代码片段和/或数据结构), 以便后续的复用。具有这种特点的服务可以潜在的解决通过服务重组来满

图 16 灵活的服务协同机制

Figure 16 Flexible service coordination mechanisms

足新的业务需求. 当业务需求有新变化时, 用户可以调整 Web 服务组合逻辑或任何其他的工作流语言指定的业务流程, 以快速响应的需求.

4 系统部署示范和实验评估

4.1 基于 EDSOA 的区域集中供热监控和信息服务系统

为了验证本文所提出的基于 EDSOA 的物联网服务提供机制, 课题组于 2012 年开发并在北京部署了区域集中供热监控和信息服务系统 (DHCISS, district heating control and information service system), 应用背景如第 2.2 节所述. 图 17 展示了事件驱动面向服务的 DHCISS 架构, 整个系统主要由两部分组成: 本地锅炉房系统和总部系统. 本地锅炉房系统由传感器信息采集、实时监控、报警、复杂事件处理和报表系统组成; 除了上面提到这些服务系统, 总部系统还包括计费、呼叫中心、维修和救援系统. 此外, 总部系统还需要与区域供热办公室的信息系统交互, 以报告突发事件和统计报表数据. 锅炉房系统的不同子系统和总部系统都通过基于虚拟专网的统一消息空间的事件代理相连接. 所有本地锅炉房系统通过不同的接入方式 (如 ADSL, GRPS, WiFi 等) 和总部系统连接到统一消息空间. 通常, 本地锅炉房系统利用一个事件代理来处理不同子系统间的事件分发. 同时, 该代理作为一个委托代理, 还负责与总部系统的事件分发. 由于总部系统需要与几十个锅炉房系统交互, 总部系统上部署了多个事件代理, 以负载均衡的集群方式分发事件. 因为无论是锅炉房系统还是总部系统都以事件驱动的方式彼此交互, 系统的耦合度因此大大降低. 图 18 展示了 DHCISS 应用系统的部分系统界面.

4.2 功能评估

为了说明事件驱动、面向服务的物联网服务的提供机制, 图 19 展示了一个简化的 DHCISS 服务场景. 该场景涉及跨业务域的多个服务动态协同: 传感器信息采集 (SIC) 服务、实时监控人机交互服务、复杂事件分析 (EA) 服务、报警服务、呼叫中心服务、维修及救援服务和紧急事件处理流程服务.

图 17 事件驱动、面向服务的 DHCIS 架构

Figure 17 EDSOA-based DHCIS architecture

图 18 部署的 DHCIS 系统界面

Figure 18 Snapshots of the deployed DHCIS system

传感器信息采集服务采集供热相关的传感器数据, 如供水温度、回水温度、管道压力和回水量等. 复杂事件分析服务接收各种实时数据流, 然后分析和聚合这些数据以找出潜在的系统问题. 实时监控人机交互服务实时显示感知数据和异常报警, 值班人员用来监控锅炉房各锅炉及换热站等物理设备的当

图 19 一个简化的事件驱动 DHCIS 服务协同场景

Figure 19 A simplified event-driven DHCIS service coordination scenario

前状态并进行相关的控制. 根据异常等级, 报警服务将以不同方式, 如光、声响、语音提示、电子邮件、短信等通知相关人员. 锅炉房系统的这些感知和报警信息都会分发给总部系统. 当总部系统的复杂事件分析服务判断这是一个高层事件时, 它会通知其他服务系统的相关服务. 例如, 当维修和救援服务接收到这个紧急维修事件时, 值班经理将组织和派遣维修人员尽快处理. 更严重的事件发生时, 呼叫中心服务需要通知受影响的用户做好相关准备. 同时, 在紧急情况下, 应当上报政府供热办的信息系统从而采取更高层面的应对措施.

事件驱动的多服务协同过程如下所示:

(1) SIC 服务 (S1) 发布的感知数据 (主题 t1) 分别被本地锅炉房系统和总部系统的 HMI 服务 (S3, S5) 接收. HMI 服务用图片和表格的形式显示感知数据给值班人员.

(2) SIC 服务 (S1) 发布的异常报警 (主题 t2) 被 HMI (S3, S5)、复杂事件分析服务 (S2)、报警服务 (S4, S6) 接收. HMI 服务用图片和表格的形式显示报警给值班人员. 同时, 报警服务将以不同方式, 如光、音响、语音、电子邮件、短信等通知负责人. 事件分析服务将分析事件, 以评估设备的安全状态.

(3) SIC 服务 (S1) 接着报告其他类型的异常报警 (主题 t3, t4). 类似步骤 (2), 这些报警都分发给 HMI (S3, S5)、事件分析服务 (S2) 和报警服务 (S4, S6).

(4) 当简单事件 t2, t3, t4 在一定的时间窗口内先后发生时, 本地锅炉房系统的事件分析服务 (S2) 将其聚合为一个高层报警事件 (主题 t5), 该事件也将分发给 HMI (S3, S5)、总部系统的事件分析服务 (S7) 和报警服务 (S4, S6).

(5) 与此同时, 在总部系统中, 呼叫中心服务 (S8) 收到了人工报警电话, 坐席人员记录人工报警信

图 20 基于 BPEL 的传统的集中式“请求—响应”告警处置服务流程

Figure 20 Centralized request-response warning disposal BPEL process

息, 然后呼叫中心服务 (S8) 发布一个问题报告报警事件 (主题 t6).

(6) 当总部系统的复杂事件分析服务 (S7) 在主题 t5 之后接收到问题报告报警事件 (主题 t6), 它判断情况是严重的, 并发布一个更严重的突发事件 (主题 t7), 统一消息空间将该事件 (主题 t7) 快速分发给呼叫中心服务 (S8)、维修及救援服务 (S9) 和紧急事件处理服务 (S10).

(7) 当接收到紧急事件 (主题 t7) 时, 呼叫中心服务 (S8) 将通过查询客户管理系统来通知受影响的用户做好相关预防准备. 维修和救援服务 (S9) 的值班经理将迅速组织和派遣维修人员来抢修. 同时, 所在管辖区域的政府供热办公室也将启动应急响应流程.

从上面的服务协同过程可以看出, 系统没有明确的定义集中式的服务编排流程, 而是基于发布/订阅的事件异步通信机制, 采用隐含的事件链来组织跨业务域甚至跨组织的异构服务系统协同, 如服务 S1 → 主题 t2 → S2 | S3 | S4 | S5 | S6. 当服务协同需求改变时, 用户只需要改变事件的发布/订阅就可以. 这样, 集成了多个异构服务系统的物联网服务系统的敏捷性和灵活性大大提高了.

4.3 性能评估

本文以 DHICSS 系统中告警处置流程为例, 对基于 EDSOA 的物联网服务协同和传统的基于“请求—响应”的集中式 SOA 服务协同的性能进行了实验比较. 利用 LoadUI⁷⁾作为性能测试工具, 采用 Apache ODE⁸⁾作为集中式 Web 服务编排引擎, 利用 BPEL 语言编制了一个基于“请求—响应”的集中式的告警处置服务流程作为测试用例, 包括了 5 个流程分支和 9 个 Web 服务 (如图 20 所示); 对于事件驱动的服务协同, 我们使用部署于实验室内部的 DHICSS 原型系统. 传感器数据由 LoadUI 脚本模拟, 采集的原始数据包括供水温度、回水温度、供水压力和室外温度. 本文主要关注在限定的时间内 (2 分钟) 两种不同服务协同模式下系统的每秒的并发事务处理数和响应时间.

如图 21 所示, 对于传统的集中式“请求—响应”服务协同模式, 可以看出, 最大的每秒事务处理

7) LoadUI. <http://sourceforge.net/projects/loadui/>

8) Apache ODE. <http://ode.apache.org/>

图 21 每秒事务处理数的比较

Figure 21 Comparison of transactions per second

图 22 服务协同的响应时间比较

Figure 22 Comparison of response time

数是 74 次, 平均的每秒事务处理数为 64.42 次. 而对于基于 EDSOA 的服务协同模式, 最大的每秒事务处理数是 167.6, 平均的每秒事务处理数是 130.4. 从图 22 中可见, 对于传统的集中式“请求—响应”服务协同模式, 最大的响应时间是 150 毫秒, 平均响应时间是 134 毫秒; 而对于基于 EDSOA 的服务协同模式, 最大的响应时间是 42 毫秒, 平均的响应时间为 30 毫秒. 由此可见, 传统的“请求—响应”的 SOA 集中式服务协同的处理速度低于事件驱动的服务协同模式, 原因就在于存在一个用于处理“请求—响应”式 BPEL 流程的中央控制点, 这会导致所有消息沿较长路径传播并等待中心队列的处理. 而事件驱动面向服务的模式通过事件的异步通信机制, 可以支持多服务的并发协同执行, 因此, 更适合时延敏感、实时性要求较高的物联网应用场景.

5 讨论及未来工作

事件驱动、面向服务的软件架构是一种更加敏捷、灵活、松耦合的服务提供机制, 非常适合于物

联网服务面向三元问题域的新特性. 然而, 为了提供更加智能和无人值守的自动化物联网应用, 还存在一定的局限性和尚待解决的问题. 在物联网环境中, 各种实体和设备主动或被动地产生大量多源的、不确定的和异构的感知信息, 如何有效地实现感知信息的实时高效融合和高层上下文信息的挖掘, 还需要针对不同的应用领域进行特定算法的深入研究, 有必要进一步探讨自动解释复杂事件, 甚至预测未来会发生什么以及应采取什么预防措施的方法. 此外, 大部分的物联网应用同时包含了大量结构化和非结构化的感知数据 (如图片、视频、RFID 数据等), 如何对这些海量异构数据进行实时高效的关联处理成为迫在眉睫的问题, 算法的时间和空间复杂性, 尤其是前者, 获得了空前重视. 一个高效的算法, 要么复杂性很低, 要么能够很好的并行化处理. 因此, 物联网服务提供应充分考虑和云计算、大数据研究的深度有机融合. 这些问题的解决将进一步促进更智能、反应更敏捷的物联网服务的出现.

另一项需要进行的工作是增强服务描述的语义互操作性. 本文目前只考虑了对物联网资源的语义建模, 还没考虑过服务描述中的语义标注问题, 这种服务提供模式更适合于确定实体、设备和企业应用系统间的服务协同. 对于高度动态、完全自组织的服务交互模式 (如车联网中道路上车辆之间的动态交互), 服务提供者和服务请求者有意义的相互沟通就变得非常必要. 为支持机-机和物-物之间无缝的语义互操作^[28], 设备和应用系统必须理解服务的含义. 虽然前期已有很多服务语义标注的研究, 但现有标准或方法的最初设计初衷主要是针对描述重量级的企业服务. 事实上, 对于大量的资源受限的嵌入式设备, 也需要考虑针对轻量级服务描述协议的语义标注问题.

6 总结

物联网的目标是实现物理世界和信息空间的无缝集成. 随着物理世界的引入, 动态的实体交互、海量的感知信息、个性化的服务需求对现有的服务提供机制提出了新的挑战. 物联网服务提供本质上是“用户域、信息空间域和物理空间域”的三元问题域的无缝协同, 扩展了现有服务系统的边界. 由于现有的服务提供机制基本上没有充分的考虑大规模感知信息的分布式处理以及如何根据物理世界实时发生的事件来跨业务域甚至跨组织的有效协调不同的企业服务流程. 因此, 如何在一个开放的分布式计算环境中实时有效地按需分发、聚合感知信息, 并且实现跨业务域或异构系统的服务动态协同, 成为物联网服务提供面临的新挑战. 通过与现有的服务提供模式比较, 本文分析了物联网服务提供面临的新的技术挑战, 通过集成 SOA 和 EDA 的互补优势, 系统化地提出了一种事件驱动、面向服务的物联网服务提供方法. 采用 SOA 技术, 解决了分布式、松耦合的物联网环境中不同异构服务间的互操作性和可复用性; EDA 实现了感知信息的按需分发和事件驱动服务的动态协同. 最后, 通过一个已部署的区域供热控制和信息服务系统 (DHCISS) 的对所提方法进行了实证研究.

参考文献

- 1 ITU. ITU Internet Reports 2005: The Internet of Things. <http://www.itu.int/osg/spu/publications/internetofthings/index.html>
- 2 Ma H D. Internet of Things: Objectives and scientific challenges. *J Comput Sci Technol*, 2011, 26: 919-924
- 3 Fok C L, Roman G C, Lu C Y. Agilla: A mobile agent middleware for self-adaptive wireless sensor networks. *ACM Trans Auton Adap*, 2009, 4: 1-26
- 4 Wang M M, Cao J N, Li J, et al. Middleware for wireless sensor networks: A survey. *J Comput Sci Technol*, 2008, 23:305-326

- 5 Bandyopadhyay S, Sengupta M, Maiti S, et al. A survey of middleware for Internet of Things: Recent Trends in Wireless and Mobile Networks, Berlin: Springer Heidelberg. 2011. 162: 288–296
- 6 Guinard D, Trifa V, Karnouskos S, et al. Interacting with the SOA-based Internet of Things: discovery, query, selection, and on-demand provisioning of web services. *IEEE Trans Serv Comput*, 2010. 3: 223–235
- 7 Motwani R, Motwani M, Harris F, et al. Towards a scalable and interoperable global environmental sensor network using service oriented architecture. In: *Proceedings of Sixth International Conference on Intelligent Sensors*. Brisbane, 2010. 151–156
- 8 Teixeira T, Hachem S, Issarny V, et al. Service oriented middleware for the Internet of Things: A perspective. In: *Proceedings of ServiceWave*. Poznan, 2011. 220–229
- 9 Wei Q, Jin Z. Service discovery for Internet of Things: a context-awareness perspective. In: *Proceedings of the Fourth Asia-Pacific Symposium on Internetware*. QingDao, 2012
- 10 Wei Q, Jin Z, Li G, et al. Preliminary study of service discovery in Internet of Things: Feasibility and limitation of SOA. *J Front Comput Sci Technol*, 2013. 7: 97–113 [魏强, 金芝, 李戈, 等. 物联网服务发现初探: 传统 SOA 的可行性和局限性. *计算机科学与探索* 2013, 7: 97–113]
- 11 Priyantha N B, Kansal A, Goraczko M, et al. Tiny web services: design and implementation of interoperable and evolvable sensor networks. In: *Proceedings of Sixth ACM Conference on Embedded network sensor systems*. New York: ACM, 2008. 253–266
- 12 Castellani A P, Gheda M, Bui N, et al. Web services for the Internet of Things through CoAP and EXI. In: *Proceedings of IEEE International Conference on Communications Workshops*. Kyoto, 2011. 1–6
- 13 Meyer S, Ruppen A, Magerkurth C. Internet of Things-Aware process modeling: integrating IoT devices as business process resources. In: *Proceedings of Advanced Information Systems Engineering*. Valencia, 2013. 7908: 84–98
- 14 Zorzi M, Gluhak A, Lange S, et al. From today's intranet of things to a future Internet of Things: a wireless and mobility-related view. *IEEE Wirel Commun*, 2010, 17: 44–51
- 15 Atzori L, Iera A, Morabito G. The Internet of Things: A survey. *Comput Netw*, 2010, 54: 2787–2805
- 16 Erl T. *Service-Oriented Architecture: Concepts, Technology, and Design*. New Jersey: Prentice Hall, 2005
- 17 Michelson B M. *Event-driven architecture overview*. Patricia Seybold Group, 2006, 1–8
- 18 Hou L S, Jin Z, Wu B D. Modeling and verifying web services driven by requirements: An ontology based approach. *Sci China Ser-F Inf Sci*, 2006, 49: 792–820
- 19 Liu X Z, Huang G, Mei H. A community-centric approach to automated service composition. *Sci China Inf Sci*, 2010, 53: 50–63
- 20 Wang P W, Jin Z, Liu H Y. Capability description and discovery of internetware entity. *Sci China Inf Sci*, 2010, 53: 685–703
- 21 Wang P W, Jin Z, Liu L, et al. Building towards capability specifications of web services based on an environment ontology. *IEEE Trans Knowl Data Eng*, 2008, 20: 547–561
- 22 Overbeek S, Klievink B, Janssen M. A flexible, event-driven, service-oriented architecture for orchestrating service delivery. *IEEE Intell Syst*, 2009, 24: 31–41
- 23 Dasgupta S, Bhat S, Lee Y. An abstraction framework for service composition in event-driven SOA systems. In: *Proceedings of IEEE International Conference on Web Services*. Los Angeles, 2009. 671–678
- 24 Zhu D, Zhang Y, Cheng B, et al. Towards a flexible event-driven SOA based approach for collaborating interactive business processes. In: *Proceedings of IEEE International Conference on Services Computing*. Washington, 2011. 749–750
- 25 Shi R S, Liu F Q, Zhang Y, et al. MID-based load balancing approach for topic-based pub-sub overlay construction. *Tsinghua Sci Technol*. 2011, 16:589–600
- 26 Zhao S, Zhang Y, Chen J L. An ontology-based IoT resource model for resources evolution and reverse evolution. In: *Proceedings of 10th International Conference on Service Oriented Computing*. Shanghai, 2012. 7636: 779–789
- 27 Villalonga C, Bauer M, López Aguilar F, et al. A resource model for the real world internet. *Smart Sens Context*, 2010, 6446: 163–176
- 28 Barnaghi P, Wang W, Henson C, et al. Semantics for the Internet of Things: early progress and back to the future. *Int J Semant Web Inf Syst*, 2012, 8: 1–21

An EDSOA-based services provisioning approach for Internet of Things

QIAO XiuQuan^{1*}, ZHANG Yang¹, WU BuDan¹, CHENG Bo¹, ZHAO Shuai¹, MA HuaDong² & CHEN JunLiang¹

1 State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China;

2 Beijing Key Lab. of Intelligent Telecommunications Software and Multimedia, School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China

*E-mail: qiaoxq@bupt.edu.cn

Abstract Internet of Things (IoT) seamlessly integrates the user domain, cyberspace and physical space domains, and realizes the dynamic human-machine-thing collaborations. In general, IoT applications often comprise the interactions of human activities, software services and physical entities spanning multiple service domains or organizations. An IoT application system often involves the aggregation of several distributed enterprise application systems. It builds a larger, composite system to collaboratively achieve a specific business goal. Most previous work only focuses on the interconnection and interoperability problems among different entities and enterprise application systems in a dynamic interaction environment. However, the problems of how to efficiently disseminate the sensing information among the information providers and consumers on-demand and how to dynamically coordinate the relevant enterprise services or business processes based on the occurred events from the real world still have not been comprehensively explored. To resolve these problems, we propose an EDSOA-based (Event-driven Service-oriented Architecture) IoT service provisioning approach which well integrates the advantages of EDA (Event-driven Architecture) and SOA (Service-oriented Architecture) in this paper. The combination of SOA and EDA can easily support the on-demand dissemination of sensing information and realize the event-driven dynamic service coordination cross domain. This proposed approach enhances the flexibility and responsiveness of IoT service systems, which allows these systems to respond dynamically, in real time, as events occur. Finally, we demonstrate its practicability by a deployed District Heating Control and Information Service System (DHCIS) application.

Keywords Internet of Things, service provisioning, event dissemination, service coordination, service-oriented architecture, event-driven architecture, complex event processing

QIAO XiuQuan received the Ph.D. degree in 2007 from School of Computer Science and Technology, Beijing University of Posts and Telecommunications (BUPT), Beijing. Now, he is an associate professor of BUPT and deputy director of Network Service Foundation Research Center of State Key Laboratory of Networking and Switching Technology. His main research interest includes services computing, Internet of Things and mobile social networking services.

MA HuaDong received his Ph.D. degree in computer science from Institute of Computing Technology, the Chinese Academy of Sciences in 1995. He is executive dean of School of Computer Science, Beijing University of Posts and Telecommunications (BUPT). He is also director of Beijing Key Lab of Intelligent Telecommunications Software and Multimedia, BUPT. He is chief scientist of the project "Basic Research on the Architecture of Internet of Things" sponsored by the National 973 Program of China. He was awarded National Funds for Distinguished Young Scientists in 2009. His current research focuses on multimedia system and networking, sensor networks and Internet of things. He is a member of the CCF, ACM, and IEEE.

ZHANG Yang received the Ph.D. degree in computer applied technology from Institute of Software, the Chinese Academy of Sciences in 2007. His research interest includes security and privacy of anonymous systems. His published papers concern anonymous routing protocols, anonymous authentication protocols, design and implementation of anonymous systems, and pseudonym systems. He currently works in State Key Laboratory of Networking and Switching Technology, Beijing University of Posts & Telecommunications, Beijing.

CHEN JunLiang graduated from the Shanghai Jiaotong University in 1955 and obtained Doctor of Engineering degree from Moscow Telecommunications Engineering Institute in 1961. Currently, he is a professor of Beijing University of Posts and Telecommunication. He was elected as a member of Academician of the Chinese Academy of Sciences in 1991 and a member of the Chinese Academy of Engineering in 1994. He is the head of the Technical Committee of Service Computing under China Computer Federation. His research interest includes communication network and communication software, services computing and Web service.